

REMARKS

Claims 1-81 are pending. No claims are allowed.

The First 35 U.S.C. § 102 Rejection

Claims 1-15, 28-32 and 55-59 stand rejected under 35 U.S.C. § 102(e) as being allegedly anticipated by McIntyre.¹ This rejection is respectfully traversed.

According to the M.P.E.P., a claim is anticipated under 35 U.S.C. § 102(a), (b) and (e) only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.² “The identical invention must be shown in as complete detail as contained in the claim.”³ “The elements must be arranged as required by the claim...”⁴

Contrary to the Examiner’s statement, each and every element is not found McIntyre. Furthermore, the various combinations of elements proposed by the Examiner are never arranged by McIntyre in the same manner as proposed by the Examiner or as required by the present claims.

¹ USP 6,178,546 B1.

² Manual of Patent Examining Procedure (MPEP) § 2131. See also *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

³ *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

⁴ *Id.*

Claim 1

Claim 1 recites:

A method for downloading code to a resource-constrained computer, the code being separable into at least one package having at least one referenceable item, comprising:

forming the package;

forming a mapping of the referenceable item to a corresponding token; and

providing the package and the mapping.

Regarding the preamble of claim 1, the Examiner states:

McIntyre provides a system for downloading code, see fig. 1. All computers are considered to be resource constrained in some manner; since, there is no such thing Unlimited storage.⁵

However, the term “resource-constrained” referred to in claim 1 does not serve to distinguish between a device having resource constraints and a device having no resource constraints. Rather, in light of the claim language, specification, and drawings of the present invention, a “resource-constrained” device is a device that is restricted or limited in memory or computing power or speed relative to a typical desktop computer. Examples of such resource-constrained devices include boundary scan devices, field programmable devices, pagers and cellular phones among many others.⁶

Claim 1 also recites in part:

forming the package;

forming a mapping of the referenceable item to a corresponding token; and

⁵ Office Action dated April 10, 2002, ¶ 2.

The Examiner states:

McIntyre's system also Provides for packaging data, see the summary of the invention, In reference to the referenceable item, see the headers, which are used to identify the data to respective targets, col. 2 lines 20-65. There is an inherent mapping between the Headers and the respective target entries, As indicated above (col. 2 lines 20-65). Further more, there is a mapping of values (Referenceable items, col. 1 lines 23-24) To tokens. McIntyre further defines Tokens and values in col. 3 lines 43-col. 4 line 65. Mapping again is inherent in Token and target entries specifically Indicated in col. 3 lines 59-65. Furthermore, note that the token specifies the type of target to be built, col. 4 line 12-20 and col. 4 lines 48-65 also lists Tokens with referenceable items. Also, it was standard in the art at the time of The invention to create separate files to perform separate functions and link the Files as needed. This feature has been specifically indicated via header files and Packaging is synonymous with mapping.⁷

Contrary to the Examiner's statement, packaging is not synonymous with mapping. The Examiner fails to comprehend the distinction between forming a map and forming a mapping. The McIntyre reference discloses a description file that contains declarations of tokens global to the rest of the file and target declaration entries for the current directory.⁸ Tokens are of the form: "token>value", where both the token name and the token value appear side-by-side in the same file.⁹ Target entries are used to describe items to be built, packaged or installed by a make generation tool.¹⁰ The following example from the McIntyre reference is illustrative.

It may be seen that the token TARGET> specifies the type of item to be built, packaged or installed. Valid target types include executable programs, variables, libraries, object files, include files, install scripts, data files, configuration files, manual pages, and help files. The NAME> token allows the developer to provide the name of the item. The SRCS> token provides the source file names. The OBJS> provides the object file names. Each target ends with an END> token.¹¹

⁶ Specification at p. 6 lines 11-26; p. 12 lines 18-19; p. 44 lines 8-18.

⁷ Office Action ¶ 2.

⁸ McIntyre at col. 2 lines 20-22.

⁹ *Id.* at col. 2 lines 23-24, col. 4 lines 5-10, 30-35 and 47-64, col. 5 lines 8-28 and 41-59.

¹⁰ *Id.* at col. 2 lines 25-26

Thus, McIntyre discloses using tokens and their associated token values to specify *how* to package target entries. Nowhere in McIntyre does it disclose *forming a mapping of the referenceable item to a corresponding token*.

Claim 1 also recites in part:

providing the package and the mapping.

The Examiner contends that McIntyre discloses providing the package and the mapping.¹² The Applicants respectfully disagree. As mentioned above, McIntyre does not disclose forming a *mapping* of the referenceable item to a corresponding token. Accordingly, McIntyre does not disclose providing the package *and the mapping*.

Since these elements are not disclosed in McIntyre, the rejection is unsupported by the art and should be withdrawn.

Claim 2

Claim 2 recites:

A method for linking code downloaded to a resource-constrained computer, the code being separable into at least one package having at least one referenceable item, comprising:
receiving the package;
receiving a mapping of the referenceable item to a corresponding token; and
linking the package using the mapping.

¹¹ *Id.* at col. 4 lines 12-20.

¹² Office Action ¶ 2.

As mentioned above with respect to claim 1, McIntyre does not disclose providing the package *and the mapping*. Furthermore, nowhere in McIntyre does it disclose receiving a mapping of the referenceable item to a corresponding token, and nowhere in McIntyre does it disclose *linking* the package *using the mapping*. Since these elements are not disclosed in McIntyre, the rejection is unsupported by the art and should be withdrawn.

Claim 3

Claim 3 recites:

A method for linking code downloaded to a computer, the computer comprising a first package that includes a mapping of tokens to externally referenceable items, the method comprising:
receiving a second package comprising at least one reference to an item in said first package, wherein said reference is represented by one or more tokens;
and
linking said second package to said first package by resolving said one or more tokens.

As mentioned above with respect to claim 1, McIntyre does not disclose providing the package *and the mapping*. Furthermore, nowhere in McIntyre does it disclose receiving a second package comprising at least one reference to an item in said first package, wherein said reference is represented by one or more tokens, and nowhere in McIntyre does it disclose linking said second package to said first package by resolving said one or more tokens. Since these elements are not disclosed in McIntyre, the rejection is unsupported by the art and should be withdrawn.

Claim 4

Claim 4 recites:

A method for constructing an image of a first package of code on a computer, the code being separable into at least one package having at least one reference to an item in a second package of code, the method comprising:
receiving a mapping of said item to at least one corresponding token;
replacing said at least one reference with said at least one corresponding token;
and
forming said package.

As mentioned above with respect to claim 1, McIntyre does not disclose providing the package *and the mapping*. Furthermore, nowhere in McIntyre does it disclose receiving a mapping of said item to at least one corresponding token, and nowhere in McIntyre does it disclose replacing said at least one reference with said at least one corresponding token. Since these elements are not disclosed in McIntyre, the rejection is unsupported by the art and should be withdrawn.

Claim 5

Claim 5 recites:

A method for constructing an image of a first package of code comprising at least one internally referenceable item, the method comprising:
forming a mapping of said at least one internally referenceable item to an optimized numeric value;
replacing references to said at least one internally referenceable item with the corresponding numeric value; and
forming the package.

As mentioned above with respect to claim 1, McIntyre does not disclose providing the package *and the mapping*. Furthermore, nowhere in McIntyre does it

disclose forming a mapping of said at least one internally referenceable item to an *optimized numeric value* and nowhere in McIntyre does it disclose replacing references to said at least one internally referenceable item with the corresponding *numeric value*. Since these elements are not disclosed in McIntyre, the rejection is unsupported by the art and should be withdrawn.

Claims 6-15

Claims 6-10 and 11-15 depend from claims 1 and 3, respectively. The base claims being allowable, the dependent claims must also be allowable.

Claims 6 and 11

Claim 6 recites:

The method of claim 1 wherein said mapping further comprises generating one or more token types, such that tokens belonging to the same token type represent the same kind of referenceable item.

The Examiner states:

In reference to the tokens belonging to the same token types (claims 6 and 11), see the group resource file (col. 3 lines 43-58).¹³

The Applicants suggest the Examiner's attempt to equate generating token types with a group resource file is improper. Contrary to the Examiner's statement, McIntyre does not disclose generating one or more token types, such that tokens belonging to the same token type represent the same kind of referenceable item. According to McIntyre,

¹³ *Id.*

A user resource file sets values for configuration variables for a single developer and typically resides in the home directory of the developer. A group resource file, on the other hand, is typically set up for a development group so that the configuration variable assignments may be shared among developers in the group.¹⁴

Thus, group resource files are used merely to share configuration variable assignments among developers in a group.

Since the limitations of claim 6 are not disclosed in McIntyre, the rejection is unsupported by the art and should be withdrawn. Claim 11 includes limitations similar to claim 6, so the rejection is also unsupported by the art and should be withdrawn.

Claims 7 and 12

Claim 7 recites:

The method of claim 1, further comprising recording in an image of said package a mapping between said token and said referenceable item.

The arguments made above with respect to claim 2 apply here as well. Since the limitations of claim 6 are not disclosed in McIntyre, the rejection is unsupported by the art and should be withdrawn. Claim 12 includes limitations similar to claim 7, so the rejection is also unsupported by the art and should be withdrawn.

¹⁴ McIntyre at col. 3 lines 48-53.

Claims 8 and 13

Claim 8 recites:

The method of claim 1 wherein said referenceable item comprises a class and said reference comprises a package and a class token.

The Examiner contends:

The object targets comprises a class/package (col. 4 lines 37-45) and inherently tokens become class/package tokens (see claims 8 and 13).¹⁵

The Applicants respectfully disagree. The Applicants respectfully submit that such conclusory allegations regarding the alleged inherency of the teachings are improper. According to the MPEP¹⁶, an Examiner must provide a rationale or evidence tending to show inherency. The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic.¹⁷ Furthermore, "In relying upon the theory of inherency, the Examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art."¹⁸

The Applicants respectfully suggest that the Examiner has failed to provide a basis in fact and/or technical reasoning to reasonably support the determination that tokens become class/package tokens necessarily flows from the teachings of McIntyre.

¹⁵ Office Action ¶ 2.

¹⁶ MPEP § 2112.

¹⁷ *In re Rijckaert*, 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993).

Nowhere in McIntyre does it disclose that a referenceable item comprises a class and a reference comprises a package and a class token. Moreover, the word “class” does not appear in the reference provided by the Examiner.

Since the limitations of claim 8 are not disclosed in McIntyre, the rejection is unsupported by the art and should be withdrawn. Claim 13 includes limitations similar to claim 8, so the rejection is also unsupported by the art and should be withdrawn.

Claims 9-10 and 14-15

Claim 9 recites:

The method of claim 1 wherein said referenceable item comprises a field and said reference comprises a package, a class and a field token.

Claim 10 recites:

The method of claim 1 wherein said referenceable item comprises a method and said reference comprises a package, a class and a method token.

The Examiner states:

In reference to claims 9-10 and 14-15, the various tokens are considered names of respective items (package, class and fields), see again col. 3 lines 43-58.¹⁹

The Applicants respectfully disagree. The arguments made above with respect to claims 8 and 13 apply here as well. Nowhere in McIntyre does it disclose that a referenceable item comprises a method and a reference comprises a package, a class and a method

¹⁸ *Ex parte Levy*, 17 USPQ2d 1461 (Bd. Pat. App. & Inter. 1990) (emphasis in original).

¹⁹ Office Action ¶ 2.

token. Additionally, nowhere in McIntyre does it disclose that a referenceable item comprises a field and a reference comprises a package, a class and a field token. Since these elements are not disclosed in McIntyre, the rejection is also unsupported by the art and should be withdrawn

Claims 14-15 include limitations similar to claims 9-10. Since these elements are not disclosed in McIntyre, the rejection is also unsupported by the art and should be withdrawn.

Claims 28-32 and 55-59

Claims 28-32 and 55-59 include limitations similar to claims 1-5. Thus, the arguments made for claims 1-5 apply here as well and McIntyre can not be said to anticipate claims 28-32 and 55-59. Accordingly, the Applicants respectfully request the rejection of claims 28-32 and 55-59 be withdrawn.

The Second 35 U.S.C. § 102 Rejection

Claims 1-81 stand rejected under 35 U.S.C. § 102(e) as being allegedly anticipated by Kyle.²⁰ This rejection is respectfully traversed.

Again, contrary to the Examiner's statement, each and every element is not found Kyle. Furthermore, the various combinations of elements proposed by the Examiner are never arranged by Kyle in the same manner as proposed by the Examiner or as required by the present claims.

²⁰ USP 6,141,681.

Claim 1

Claim 1 recites in part:

A method for downloading code to a resource-constrained computer, the code being separable into at least one package having at least one referenceable item, comprising:

The Examiner states:

Kyle provides a system for downloading code with instructions, see fig. 2. All computers are considered constrained in some manner; since, there is no such thing as Unlimited storage or resources. Kyle also provides for packaging data, see the Summary of the invention. In reference to the referenceable item, see the instructions, which Are used to identify how the data is to be used by respective targets, see fig. 8 and col. 1 line 60-col. 2 line 53.²¹

Again, the term “resource-constrained” referred to in claim 1 does not serve to distinguish between a device having resource constraints and a device having no resource constraints. Rather, in light of the claim language, specification, and drawings of the present invention, a “resource-constrained” device is a device that is restricted or limited in memory or computing power or speed relative to a typical desktop computer. Examples of such resource-constrained devices include boundary scan devices, field programmable devices, pagers and cellular phones among many others.²²

Claim 1 also recites in part:

forming the package;
forming a mapping of the referenceable item to a corresponding token; and

The Examiner states:

²¹ Office Action ¶ 3.

see col. 3 lines 55-65, which indicates that Requested data (token) and its corresponding execution instruction (referenceable item). Note also that the items are assembled (mapped) and downloaded. Kyle also utilizes headers (Referenceable items) to identify how Data (tokens) are interpreted, col. 4 lines 33-56 and col. 4 line 66-col. 5 line 32. Also, See figs. 5 and 6, which indicates that the Packaged items are mapped together.²³

Again, the Examiner fails to comprehend the distinction between forming a map and forming a mapping. Contrary to the Examiner's statement, assembling is not synonymous with mapping, nor is assembling synonymous with forming a mapping. The Examiner equates the term "referenceable item" in claim 1 with (1) instructions which are used to identify how the data is to be used by respective targets and (2) headers used to identify how data are interpreted.

The Examiner also equates the term "token" with the data interpreted by the instructions. The applicants respectfully submit that equating the term "token" with data is improper. Kyle does it disclose using a "token". Furthermore, nowhere in Kyle does it disclose *forming a mapping* of the referenceable item to a corresponding token. Rather, Kyle discloses:

On a host computer 200, data is *combined* with instructions in a single data package 212 which is treated as an object which conveys its own behavior in a destination system, local computer 204.²⁴ (emphasis added)

Nowhere does Kyle disclose forming a mapping of the referenceable item to a corresponding token.

²² Specification at p. 6 lines 11-26; p. 12 lines 18-19; p. 44 lines 8-18.

²³ Office Action ¶ 3.

Claim 1 also recites in part:

providing the package and the mapping.

The Examiner contends that Kyle discloses providing the package and the mapping.²⁵ The Applicants respectfully disagree. As mentioned above, Kyle does not disclose forming a *mapping* of the referenceable item to a corresponding token. Accordingly, Kyle does not disclose providing the package *and the mapping*.

Since these elements are not disclosed in Kyle, the rejection is unsupported by the art and should be withdrawn.

Claim 2

Claim 2 recites:

A method for linking code downloaded to a resource-constrained computer, the code being separable into at least one package having at least one referenceable item, comprising:
receiving the package;
receiving a mapping of the referenceable item to a corresponding token; and
linking the package using the mapping.

As mentioned above with respect to claim 1, Kyle does not disclose providing the package *and the mapping*. Furthermore, nowhere in Kyle does it disclose receiving a mapping of the referenceable item to a corresponding token, and nowhere in Kyle does it disclose *linking* the package *using the mapping*. Since these elements are not disclosed in Kyle, the rejection is unsupported by the art and should be withdrawn. Since these

²⁴ Kyle at col. 3 lines 29-32.

elements are not disclosed in Kyle, the rejection is unsupported by the art and should be withdrawn.

Claim3

Claim 3 recites:

A method for linking code downloaded to a computer, the computer comprising a first package that includes a mapping of tokens to externally referenceable items, the method comprising:
 receiving a second package comprising at least one reference to an item in said first package, wherein said reference is represented by one or more tokens;
 and
 linking said second package to said first package by resolving said one or more tokens.

As mentioned above with respect to claim 1, Kyle does not disclose providing the package *and the mapping*. Furthermore, nowhere in Kyle does it disclose receiving a second package comprising at least one reference to an item in said first package, wherein said reference is represented by one or more tokens, and nowhere in Kyle does it disclose linking said second package to said first package by resolving said one or more tokens. Since these elements are not disclosed in Kyle, the rejection is unsupported by the art and should be withdrawn.

Claim 4

Claim 4 recites:

A method for constructing an image of a first package of code on a computer, the code being separable into at least one package having at least one reference to an item in a second package of code, the method comprising:

²⁵ Office Action ¶ 3.

receiving a mapping of said item to at least one corresponding token;
replacing said at least one reference with said at least one corresponding token;
and
forming said package.

As mentioned above with respect to claim 1, Kyle does not disclose providing the package *and the mapping*. Furthermore, nowhere in Kyle does it disclose receiving a mapping of said item to at least one corresponding token, and nowhere in Kyle does it disclose replacing said at least one reference with said at least one corresponding token. Since these elements are not disclosed in Kyle, the rejection is unsupported by the art and should be withdrawn.

Claim 5

Claim 5 recites:

A method for constructing an image of a first package of code comprising at least one internally referenceable item, the method comprising:
forming a mapping of said at least one internally referenceable item to an optimized numeric value;
replacing references to said at least one internally referenceable item with the corresponding numeric value; and
forming the package.

As mentioned above with respect to claim 1, Kyle does not disclose providing the package *and the mapping*. Furthermore, nowhere in Kyle does it disclose forming a mapping of said at least one internally referenceable item to an *optimized numeric value* and nowhere in Kyle does it disclose replacing references to said at least one internally referenceable item with the corresponding *numeric value*. Since these elements are not disclosed in Kyle, the rejection is unsupported by the art and should be withdrawn.

Claims 6-15

Claims 6-10 and 11-15 depend from claims 1 and 3, respectively. The base claims being allowable, the dependent claims must also be allowable.

Claims 6 and 11

Claim 6 recites:

The method of claim 1 wherein said mapping further comprises generating one or more token types, such that tokens belonging to the same token type represent the same kind of referenceable item.

The Examiner states:

In reference to the tokens belonging to the same token types (claims 6 and 11), see the various format support (col. 2 lines 17-53).

The Applicants suggest the Examiner's attempt to equate generating token types with data formats is improper. Contrary to the Examiner's statement, Kyle does not disclose *generating* one or more token types, such that tokens belonging to the same token type represent the same kind of referenceable item. Rather, Kyle discloses *combining* data and corresponding instructions in a single data package.²⁶

Since these elements are not disclosed in Kyle, the rejection is unsupported by the art and should be withdrawn. Claim 11 includes limitations similar to claim 6, so the rejection is also unsupported by the art and should be withdrawn.

²⁶ Kyle at col. 3 lines 27-34.

Claims 7 and 12

Claim 7 recites:

The method of claim 1, further comprising recording in an image of said package a mapping between said token and said referenceable item.

The arguments made above with respect to claim 2 apply here as well. Since the limitations of claim 6 are not disclosed in Kyle, the rejection is unsupported by the art and should be withdrawn. Claim 12 includes limitations similar to claim 7, so the rejection is also unsupported by the art and should be withdrawn.

Claims 8 and 13

Claim 8 recites:

The method of claim 1 wherein said referenceable item comprises a class and said reference comprises a package and a class token.

The Examiner contends:

The object targets comprises a class/package (col. 3 lines 27-39, via objects/packages) and inherently tokens become class/package tokens (see claims 8 and 13).²⁷

The Applicants respectfully suggest that the Examiner has failed to provide a basis in fact and/or technical reasoning to reasonably support the determination that tokens become class/package tokens necessarily flows from the teachings of Kyle.

²⁷ Office Action ¶ 3.

Nowhere in Kyle does it disclose that a referenceable item comprises a class and a reference comprises a package and a class token. Moreover, the word “class” does not appear in the reference provided by the Examiner.

Since these elements are not disclosed in Kyle, the rejection is unsupported by the art and should be withdrawn. Claim 13 includes limitations similar to claim 7, so the rejection is also unsupported by the art and should be withdrawn.

Claims 9-10 and 14-15

Claim 9 recites:

The method of claim 1 wherein said referenceable item comprises a field and said reference comprises a package, a class and a field token.

Claim 10 recites:

The method of claim 1 wherein said referenceable item comprises a method and said reference comprises a package, a class and a method token.

The Examiner states:

In reference to claims 9-10 and 14-15, the various tokens are considered names of respective items (package, class and fields), see again col. 3 lines 27-39 and lines 56-65.²⁸

The Applicants respectfully disagree. The arguments made above with respect to claims 8 and 13 apply here as well. Claims 14-15 include limitations similar to claims 9-10.

²⁸ *Id.*

Since these elements are not disclosed in Kyle, the rejection is unsupported by the art and should be withdrawn.

Claim 16

Claim 16 recites:

The method of claim 1 wherein said package further comprises interfaces and interface method definitions; and said method further comprises constructing at least one interface method table for a class.

The Examiner contends:

The features of claims 16 are taught via Kyle's interpreting (translating, col. 7 line 53-col. 8 line 19) function, which indicates what item names could be and what the items could represent (libraries, programs or help files).²⁹

The Applicants respectfully disagree. Nowhere in Kyle does it disclose a package comprising interfaces and interface method definitions, and nowhere in Kyle does it disclose constructing at least one interface method table for a class.

Since the limitations of claim 16 are not disclosed in Kyle, the rejection is unsupported by the art and should be withdrawn.

Claims 17-27

Claims 17-19, 20-21, 22-27 depend from claims 1, 3 and 5, respectively. The base claims being allowable, the dependent claims must also be allowable.

²⁹ *Id.*

Claims 17-27 recite additional limitations not disclosed by Kyle. Claims 17-27

recite:

17. The method of claim 16 wherein said constructing comprises:
obtaining said interfaces;
constructing an ordered table of methods for each interface; and
recording an indication of the implementation of the interface method for each table entry.

18. The method of claim 17 wherein said ordered interface method table entries correspond to token values assigned to interface methods within the scope of said class.

19. The method of claim 18 wherein said indication of said implementation of said interface method comprises an index into a virtual method table.

20. The method of claim 3, further comprising resolving interface method references during execution using interface method tables, said resolving comprising:
obtaining an associated instance;
obtaining a class description of said instance;
locating an interface table of said interface method in said class description;
locating an interface method entry within an interface table of a referenced method; and
locating the implementation of said interface method based on the table entry content.

21. The method of claim 20 wherein an indication of said implementation of said interface method comprises an index into a virtual method table.

22. The method of claim 5 wherein said package further comprises at least one reference to an internal item.

23. The method of claim 22 wherein said internal item comprises a class and said reference comprises an offset within said package to a class record associated with said class.

24. The method of claim 22 wherein said internal item comprises a static field and said reference comprises an offset within said package to a value for said static field.

25. The method of claim 22 wherein said internal item comprises a static method and said reference comprises an offset within said package to code associated with said static method.

26. The method of claim 22 wherein said internal item comprises an instance field and said reference comprises an offset within said package to a class record of said instance field and a field token.

27. The method of claim 22 wherein said internal item comprises a virtual method and said reference comprises an offset within said package to a class record of said virtual method and a method token.

The Examiner contends:

The features of claims 17-27 are considered standard to object oriented programming and inherent to Kyle's system (col. 3 lines 27-35).³⁰

The Applicants respectfully disagree. The Applicants respectfully suggest that the Examiner has failed to provide a basis in fact and/or technical reasoning to reasonably support the determination that features of claims 17-27 necessarily flows from the teachings of Kyle.

Since the limitations of claims 17-27 are not disclosed in Kyle, the rejection of claims 17-27 is unsupported by the art and should be withdrawn.

Claims 28-54 and 55-81

Claims 28-54 and 55-81 include limitations similar to claims 1-27. Thus, the arguments made for claims 1-5 apply here as well and Kyle can not be said to anticipate claims 28-54 and 55-81. Accordingly, the Applicants respectfully request the rejection of claims 28-54 and 55-81 be withdrawn.

³⁰ *Id.*

The Third 35 U.S.C. § 102 Rejection

Claims 1-2, 28-32 and 55-59 stand rejected under 35 U.S.C. § 102(e) as being allegedly anticipated by Houha et al.³¹ This rejection is respectfully traversed.

Again, contrary to the Examiner's statement, each and every element is not found Houha et al. Furthermore, the various combinations of elements proposed by the Examiner are never arranged by Houha et al. in the same manner as proposed by the Examiner or as required by the present claims.

Claim 1

Claim 1 recites in part:

A method for downloading code to a resource constrained computer, the code being separable into at least one package having at least one referenceable item

The Examiner states:

Houha provides a system for downloading code with linking provided at the destination computers via an inherent referenceable item. This inherently occurs in a constrained area As Unlimited storage or resources are non Existent. Houha provides for packaging data And providing a means for linking (Mapping) appropriate data, see the summary of the invention. In reference to the referenceable item, see the headers. which Are used to identify how the data is to be used by respective targets, see The abstract of the invention.³²

The Applicants respectfully disagree.

Again, the term "resource-constrained" referred to in claim 1 does not serve to distinguish between a device having resource constraints and a device having no resource constraints. Rather, in light of the claim language, specification, and drawings of the present invention, a "resource-constrained" device is a device that is restricted or limited

³¹ USP 5,734,822.

in memory or computing power or speed relative to a typical desktop computer.

Examples of such resource-constrained devices include boundary scan devices, field programmable devices, pagers and cellular phones among many others.³³

Claim 1 also recites in part:

forming the package;
forming a mapping of the referenceable item to a corresponding token; and

The Examiner states:

see col. Fig. 3 for the mapping feature and Note again that headers are inherently mapped to file names (tokens).³⁴

The Applicants respectfully disagree. The Applicants respectfully suggest that the Examiner has failed to provide a basis in fact and/or technical reasoning to reasonably support the determination that mapping headers to file names necessarily flows from the teachings of Houha et al. Rather, Houha et al. states:

For example, header 401 may *include* a name 401a indicating the name of the *load module*...³⁵ (emphasis added)

The Examiner equates the term “referenceable item” in claim 1 with a header as disclosed in Houha et al. The Examiner also improperly equates the term “token” in claim 1 with a file name as disclosed in Houha et al. In support of the contention that Houha et al. discloses the “forming a mapping of the referenceable item to a corresponding token” element, the Examiner refers to FIG. 3. However, FIG. 3 illustrates a dispatch table mapper that includes an association between symbols and table entries.

³³ Specification at p. 6 lines 11-26; p. 12 lines 18-19; p. 44 lines 8-18.

³⁴ Office Action ¶ 4.

³⁵ Houha et al. at col. 6 lines 52-54.

It does not disclose forming a mapping of a referenceable item to a corresponding token, and it does not disclose forming a mapping of a header to a file name as suggested by the Examiner.

Claim 1 also recites in part:

providing the package and the mapping.

The Examiner contends that Houha discloses providing the package and the mapping.³⁶ The Applicants respectfully disagree. In support of this contention, the Examiner cites portions of Houha et al. that speak generally about partitioning functions normally performed by a linking loader into those that can be done at the downloading source, and those that must be performed by a loader in the terminal.³⁷ It does not disclose providing the package and the mapping of the referenceable item (header) to a corresponding token (file name) as suggested by the Examiner.

Claim 2

Claim 2 recites:

A method for linking code downloaded to a resource-constrained computer, the code being separable into at least one package having at least one referenceable item, comprising:
receiving the package;
receiving a mapping of the referenceable item to a corresponding token; and
linking the package using the mapping.

³⁶ Office Action ¶ 4.

³⁷ Houha et al., at col. 2 line 32 – col. 3 line 12.

As mentioned above with respect to claim 1, Houha et al. does not disclose providing the package *and the mapping*. Furthermore, nowhere in Houha et al. does it disclose receiving a mapping of the referenceable item to a corresponding token, and nowhere in Houha et al. does it disclose *linking* the package *using the mapping*. Since these elements are not disclosed in Houha et al., the rejection is unsupported by the art and should be withdrawn. Since these elements are not disclosed in Houha et al., the rejection is unsupported by the art and should be withdrawn.

Claim3

Claim 3 recites:

A method for linking code downloaded to a computer, the computer comprising a first package that includes a mapping of tokens to externally referenceable items, the method comprising:
receiving a second package comprising at least one reference to an item in said first package, wherein said reference is represented by one or more tokens;
and
linking said second package to said first package by resolving said one or more tokens.

As mentioned above with respect to claim 1, Houha et al. does not disclose providing the package *and the mapping*. Furthermore, nowhere in Houha et al. does it disclose receiving a second package comprising at least one reference to an item in said first package, wherein said reference is represented by one or more tokens, and nowhere in Houha et al. does it disclose linking said second package to said first package by resolving said one or more tokens. Since these elements are not disclosed in Houha et al., the rejection is unsupported by the art and should be withdrawn.

Claim 4

Claim 4 recites:

A method for constructing an image of a first package of code on a computer, the code being separable into at least one package having at least one reference to an item in a second package of code, the method comprising:
receiving a mapping of said item to at least one corresponding token;
replacing said at least one reference with said at least one corresponding token;
and
forming said package.

As mentioned above with respect to claim 1, Houha et al. does not disclose providing the package *and the mapping*. Furthermore, nowhere in Houha et al. does it disclose receiving a mapping of said item to at least one corresponding token, and nowhere in Houha et al. does it disclose replacing said at least one reference with said at least one corresponding token. Since these elements are not disclosed in Houha et al., the rejection is unsupported by the art and should be withdrawn.

Claim 5

Claim 5 recites:

A method for constructing an image of a first package of code comprising at least one internally referenceable item, the method comprising:
forming a mapping of said at least one internally referenceable item to an optimized numeric value;
replacing references to said at least one internally referenceable item with the corresponding numeric value; and
forming the package.

As mentioned above with respect to claim 1, Houha et al. does not disclose providing the package *and the mapping*. Furthermore, nowhere in Houha et al. does it

disclose forming a mapping of said at least one internally referenceable item to an *optimized numeric value* and nowhere in Houha et al. does it disclose replacing references to said at least one internally referenceable item with the corresponding *numeric value*. Since these elements are not disclosed in Houha et al., the rejection is unsupported by the art and should be withdrawn.

Claims 28-32 and 55-59

Claims 28-32 and 55-59 include limitations similar to claims 1-5. Thus, the arguments made for claims 1-5 apply here as well and Houha et al. can not be said to anticipate claims 28-32 and 55-59. Accordingly, the Applicants respectfully request the rejection of claims 28-32 and 55-59 be withdrawn.

In view of the foregoing, it is respectfully asserted that the claims are now in condition for allowance.

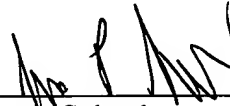
Request for Allowance

It is believed that this Amendment places the above-identified patent application into condition for allowance. Early favorable consideration of this Amendment is earnestly solicited.

If, in the opinion of the Examiner, an interview would expedite the prosecution of this application, the Examiner is invited to call the undersigned attorney at the number indicated below.

Respectfully submitted,
Thelen Reid and Priest LLP

Dated: September 9, 2002



John R. Schaub
Reg. No. 42,125

Thelen Reid and Priest LLP
P.O. Box 640640
San Jose, CA 95164-0640
(408) 282-1858